Labeling of Human Motion Based on CBGA and Probabilistic Model

[1]Fuyuan Hu, [2]Hau San Wong

[1]Suzhou University of Science and Technology, Jiangsu Prov., P.R. China

Emails: fuyuanhu@mail.usts.edu.cn

[2]City University of Hong Kong, P.R. China

Submitted: ***Dec.12, 2012*** ***Accepted: Mar. 20, 2013*** ***Published: Apr. 10, 2013***

Abstract—***In this paper, we present a novel method for the labeling of human motion which uses Constraint-Based Genetic Algorithm (CBGA) to optimize the probabilistic model of body features and construct the set of conditional independence relations among the body features by a fitness function. The approach also allows the user to add custom rules to produce valid candidate solutions to achieve more accurate results with constraint-based genetic operators. Specifically, we design the fitness function using a probability model based on the decomposable triangle model(DTM), which is learned through the EM algorithm with the minimum description length (MDL) principle and CBGA algorithm to characterize the stochastic and dynamic relations of articulated human motions. We also extend these results to learning the probabilistic structure of human body to improve the labeling results, the handling of missing body parts, the integration of multi-frame information and the accuracy rates. Finally, we analyze the performance of our proposed approach and show that it outperforms most of the current state of the art methods on a set of motion captured walking, running and dancing sequences in terms of quality and robustness.***

Index terms**: Constraint-Based Genetic Algorithm, Minimum Description Length, Labeling, Decomposable Triangle Model.**

I. INTRODUCTION

Labeling human body parts is an active topic in computer graphics and computer vision, and has many applications, including human computer interfaces, security, virtual reality, etc. Many researchers have demonstrated that activity, age and gender can be perceived easily from a series of light-dot displays, even when no other cues are available [1, 2]. Labeling human body parts is also a key pre-processing step for further analysis, such as the tracking of human [3], behavior analysis [4-5], and motion synthesis [6], etc. Many techniques have been applied to tackle these difficult problems, including the adoption of different human motion models with associated optimization algorithm [7-10].

The problem of labeling human body parts is to assign the appropriate labels for unknown body parts, and a large computational cost is in general required to find the optimal labeling. To improve the accuracy and to decrease the computational time, it is vital to select a good model with associated optimization method. Probabilistic model has been popular since realizing efficient learning and testing possibility. These methods can be classified into two main categories: tree structure and triangulated graph. Tree structured probabilistic models, which approximate a *t*-th order probability distribution by a product of second-order component distributions and search for the optimal structure by the maximum spanning tree algorithm [11], admit simple and efficient inference [12]. However, they are not well suited to the handling of problems such as occlusion and aspect variation [13], and there is no existing algorithm which runs in polynomial time and guarantees to attain the optimal solution [14]. To address the problem that multiple components of an object may be occluded simultaneously, mixtures of trees with each component modeling a particular aspect has been proposed [13] and applied to track human body [15]. A decomposable triangulated graph [16] is another type of graph to represent the human body, and is more powerful than trees since each node has two parents [17]. In previous works, Yang Song [17-19] models each triangle by a Gaussian distribution, and as a result the joint probability density function of all the parts is also a Gaussian distribution. To better express the variability and different phases of human motion, they extended the algorithms by adopting mixtures of decomposable triangulated models [20], which are mixtures of Gaussian densities with an invariant number of

components.

In the previous work, the human structures were crafted by hand to avoid exponential combinatorial searching. Currently, many techniques have been proposed for labeling human body parts and learning the body structure. Methods of finding the optimal structure may be categorized into greedy algorithm or genetic algorithm. Amit Yali [16] developed a dynamic programming algorithm to perform expansion on the graphs by adding edges to graph models, and it has been shown to be efficient when applied to the labeling of in the algorithm body parts. The greedy algorithm grows the graph according to the intrinsic property of decomposable graphs, which we start with a single vertex, and additional vertices are incorporated one in each subsequent iterations. However, the conditional independence relations are learned separately, which does not guarantee to find a global optimal network structure. Larranaga [21] used GAs to find the optimal node ordering of Bayesian networks. He represented a node ordering as a candidate solution in the population, and for each ordering, the solution is passed to K2, a greedy search algorithm, to obtain a network. In this work, the ordering and the conditional independence relations are also learnt separately, and the computation cost is very huge in this GA-based method. To resolve the computational efficiency problem of GA, search space reduction using constraints and supervised learning have been developed. Garofalakis [22] proposed a constraint-based algorithm to specify the expected tree size and accuracy in the searching process. In other words, constraints can be used as a trade-off mechanism between the model accuracy and computational efficiency in the tree-building or tree-pruning process. Chiu [23] proposed a constraint-based genetic algorithm to reveal more accurate and significant classification rules. This approach allows constraints to be specified as relationships between attributes according to predefined requirements, users' preferences, or partial knowledge in the form of a constraint network. Barnier and Brisset [24] developed a new optimization method based on a GA mixed constraint satisfaction problem(CSP) technique, whose main idea is to handle sub-domains of the CSP variables by using the GA for combinatorial problems, instead of utilizing a constraint-based reasoning mechanism to improve the GAs computation efficiency. In view of the relative advantages and disadvantages of these previous techniques, we prepare to apply constraints to effectively supervise GA evolution to reduce the searching space and improve

computation efficiency.

In this paper, we present a different approach to the problem of labeling using Gaussian mixture model with different number of components for each triangle, and constraint-based genetic algorithm to improve the labeling accuracy. Specifically, the complete set of body parts as a whole is described via a specially defined fitness function, rather than learning separately the conditional independence relations. In general, different motion types are associated with the different articulated joints. Thus, it is necessary for different triangles to adopt joint mixture Gaussian distributions with different number of components. We employ EM algorithm with the minimum description length (MDL) criterion to train the mixture Gaussian model for multivariate training data, which is capable of selecting the number of components by unsupervised learning of finite mixture models on multivariate training data [25, 26]. The genetic algorithm is then used to label the body parts by the fitness function. However, many iteration times are in general required to search for a valid candidate solution in GA, which leads to a high computational requirement when the search space is large. To alleviate this problem, we can remove the invalid candidate solutions and generate a valid solution by a supervised technique, which leads to the acceleration of the evolution process. Thus, we propose a novel GA to discover the optimized labeling of the body parts via constraint-based operation and a specially designed fitness function. We refer to our optimization algorithm as the Constraint-based Generic Algorithm (CBGA). Specifically, CBGA consists of a set of constraint-based operators and a constraint-based evaluation function. At the same time, we extend this technique to resolve the missing body parts by suitably modifying the fitness function. Finally, we integrate information from multiple frames to improve the accuracy and robustness of labeling.

The contributions of this paper are threefold: first, the proposed decomposed triangle model, which is the joint probability distribution of a Gaussian mixture model with different number of components for each triangle, is very efficient for human motion. Second, CBGA is efficient for computing the maximum-likelihood labels via the constraint-based operator and the fitness function. Third, we generate the probabilistic model using CBGA to improve the accuracy of labeling. Finally, we extend the results to handle the problem of missing body parts, and improve the correct rate of labeling by using information from multiple frames.

After reviewing related works in Section 2, we introduce the probability model for human motion representation in Section 3. The labeling algorithm based on CBGA is explained in Section 4, and experiment results are shown in Section 5. We conclude our paper in Section 6.

II. Overview of Our Approach

Let $X = (X_1, X_2, \cdots, X_N)$ be the vector of specific measurements of body features in a frame, and $L = (L_1, L_2, \cdots, L_M)$ be the vector of labels for M markers such as head, shoulder, etc. N is not always equal to M since there is the possibility of erroneous detection and/or omission of body features due to occlusion in human motion data acquisition. Here we first assume that there are no missing body parts and no clutters. That is to say N is equal to M . Then, the labeling problem is to find $\bar{L}^*$ which maximizes the posterior probability $P(\bar{L} / \bar{X})$, over all possible label vectors $\bar{X}$.

$$\bar{L}^* = \arg\max_{\bar{L} \in L} P(\bar{L} / \bar{X}) \tag{1}$$

In a maximum likelihood setting, we want to find the decomposable triangulated graph G , such that $P(G / X)$ is maximized over all possible graphs.

$$P(G / X) = P(X / G)P(G) / P(X) \tag{2}$$

We assume the priors $P(G)$ are equal for different decompositions, so our goal is to find the structure G^* which can maximize $P(X / G)$. This can be computed as follows,

$$G^* = \arg\max_{G \in \mathbf{G}} \mathrm{P(X/G)} \tag{3}$$

$$\log(P(X / G)) = \log(P(S_{body}(X_{S_1}, X_{S_2}, \cdots, X_{S_M}) / G)) \tag{4}$$

In this paper, we choose DTM to characterize the body pose and motion. In DTM, there exists a particular order of successive elimination of the cliques, so that there must be a free vertex which is only contained in each clique, and then it is eliminated. The remaining sub-graph is again a collection of cliques of three. That is to say, when a free vertex is eliminated, the next clique in the ordering will again have a free vertex to be eliminated, and so on until the last clique [16]. In [8], the authors associated a joint Gaussian mixture distribution with the same

number of components to each triangle. In contrast, we adopt a joint mixture Gaussian distribution with different number of mixture components to learn the best triangulated model, since different motion types are associated with the different articulated joints. In addition, it is capable of selecting the number of components automatically by integrating MDL into the EM algorithm. To obtain the optimal labeling of the body parts, the brute force solution is to search exhaustively among all $\prod_{i=0}^{M-1}(N-i)$ combinations, and the resulting computational cost is very large. Therefore, we propose the CBGA to obtain a more accurate labeling in an efficient way, and the framework of this approach is shown in Fig.1. The novel GA has three main components: encoding, application of genetic operators and fitness function. Encoding is a very important issue for GA. It is possible to encode the connectivity relationships of graphs using integer strings, while there are many invalid candidate solutions which do not satisfy the constraints of DTM. Thus, it is necessary to supervise the evolution process by specially designed rules to produce valid candidate solutions. As a result, constraint-based operators are introduced in this paper, which lead to a reduction of the search space and acceleration of the evolution process.

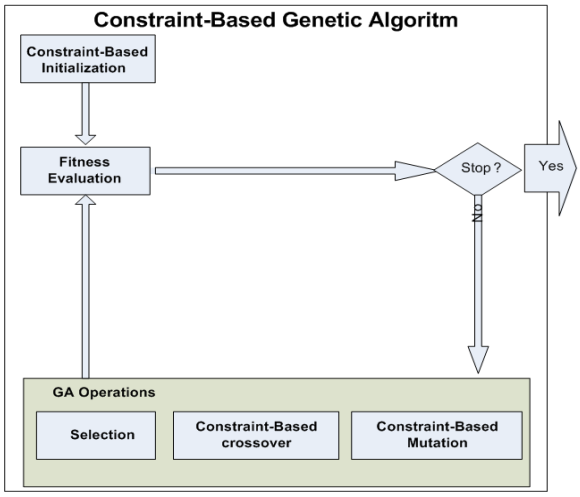

Figure1. Framework of CBGA

III. Human Models Based on Finite Mixture Models

We use DTM (decomposable triangulated model) to model the body structure, which has been used in [20] and is shown in Fig.2. In [27], DTM was used to represent the conditional

independence of body parts using a Gaussian mixture model, and we generalize this approach in this paper.

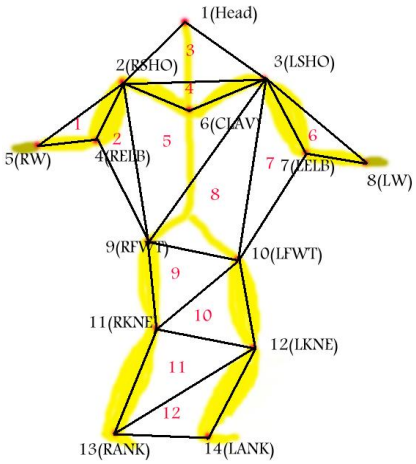

Figure2. Body parts in image plane

Given the set X of M parts and the corresponding measurements $X_{S_i}, 1 \le i \le M$, where $X = (X_{S_1}, X_{S_2}, \cdots, X_{S_N})$, we first assume N is equal to M. If the conditional independence of body parts S_{body} can be represented as a decomposable triangulated graph, the joint probability density function $P_{S_{body}}$ can be decomposed as,

$$\begin{aligned} &\log(P(X/G;\theta)) = \log(\prod_{t=1}^{T-1} P(X_{A_t} / X_{B_t}, X_{C_t};\theta_t) P(P(X_{A_T} / X_{B_T}, X_{C_T};\theta_T)) \\ &= \sum_{t=1}^{T-1} \log P(X_{A_t} / X_{B_t}, X_{C_t};\theta_t) + \log P(X_{A_T} / X_{B_T}, X_{C_T};\theta_T) \\ &= -\sum_{t=1}^{T-1} h(X_{A_t} / X_{B_t}, X_{C_t};\theta_t) - h(X_{A_T} / X_{B_T}, X_{C_T};\theta_T) \end{aligned} \tag{5}$$

where X_{A_t}, X_{B_t} and X_{C_t} are body parts, and $h(\bullet)$ denotes the entropy[28]. The optimization can be performed by maximizing equation (4). Considering the variability and different phases of human movement, we model each triangle by a Gaussian mixture distribution with a different number of components.

More formally, a K_t-component mixture model can be represented by

$$\begin{cases} G_t = [G_t^1, G_t^2, \cdots, G_t^{k_t}] \\ \omega_t = [\omega_t^1, \omega_t^2, \cdots, \omega_t^{k_t}] \end{cases} \tag{6}$$

where $G_t^i (i = 1,2,\cdots,k_t)$ is a multivariate Gaussian distribution, and ω_t^i is the prior probability

of G_t^i.

$$G_t^i = \frac{e^{[-\frac{1}{2}(X_t-\overline{X}_t^i)^T(\Sigma_t^i)^{-1}(X_t-\overline{X}_t^i)]}}{(2\pi)^{d/2}\,|\,\Sigma_t^i\,|^{\frac{1}{2}}} \tag{7}$$

Where k_t, $\overline{X}_t$, Σ_t, and ω_t can be learnt through a modified EM algorithm using the minimum description length(MDL) criterion for the t-th triangle. We specify a range from $K_{\max}$ to $K_{\min}$ for the number of components, and initialize the parameters as $\hat{\theta}(0)$. For *d*-dimensional data, each Gaussian mixture component is associated with a covariance matrix with $N = (d + d*(d+1)/2)$ parameters. The algorithm is then described below:

Inputs: $K_{\max}$, $K_{\min}$, ε, initial parameters $\hat{\theta}(0)$, data sample x.

E-Step: Calculate the conditional expectation:

$$\begin{aligned}\alpha_m^i &\equiv E[z_m^{(i)} / x, \hat{\theta}_m(t)] \\ &= \frac{\hat{\omega}_m(t)P(x^{(i)} / \hat{\theta}_m(t))}{\sum_{j=1}^{k}\hat{\omega}_j(t)P(x^{(i)} / \hat{\theta}_j(t))}\end{aligned} \tag{8}$$

Modified M-Step: Calculate the parameters,

$$\hat{\omega}_m(t+1) = \frac{\max(0, \sum_{i=1}^{n}\alpha_m^{(i)} - \frac{N}{2})}{\sum_{j=1}^{k}\max(0, \sum_{i=1}^{n}\alpha_j^{(i)} - \frac{N}{2})} \tag{9}$$

where $m = 1,2,\cdots,k$.

$$\hat{\theta}_m(t+1) = \arg\max_{\theta_m} Q(\theta, \hat{\theta}(t)) \tag{10}$$

where $\hat{\omega}_m(t+1) > 0$.

MDL Criterion: minimize the description length of the model,

$$\hat{K}_{MDL} = \arg\min_k \{-\log P(x / \hat{\theta}(k)) + \frac{k}{2}\log N\} \tag{11}$$

Output: Mixture model in $\hat{\theta}_{best}$.

Let X represents the 15-dimensional feature vector of the triplet $\Delta = \{A, B, C\}$,

$$X = (v_{A_x}, v_{B_x}, v_{C_x}, v_{A_y}, v_{B_y}, v_{C_y} v_{A_z}, v_{B_z}, v_{C_z}, p_{B_x}, p_{C_x}, p_{B_y}, p_{C_y}, p_{B_z}, p_{C_z}) \tag{12}$$

The first nine dimensions of X are *x*, *y*, *z* velocity components of the body parts (A,B,C); and the last six dimensions are positions of the body parts *B* and *C* relative to *A*. The velocity of each marker is obtained by subtracting its position in two consecutive frames. For each triangle, its probability can be obtained by:

$$P(X_{A_t} / X_{B_t}, X_{C_t}) = \max_i G_t^i (i = 1,2,\cdots,k_t) \tag{13}$$

where k_t is the number of mixture components for the *t*-th triangle.

IV. Labeling the Human Body with Constraint-Based Genetic Algorithm

For an arbitrary frame, the larger the probability value is, the more likely the labeled body parts are the right markers. If the conditional independence relationship is considered, we obtain the following,

$$\max \log P(X / G) = \max(\sum_{t=1}^{T-1} \log(X_{A_{tT}}, X_{B_t}, X_{C_t})) + \log P(X_{A_T}, X_{B_T}, X_{C_T}) \tag{14}$$

However, the number of all possible combinations is extremely large, and in view of this, we adopt CBGA to address this global optimization problem via a suitably defined fitness function. To reduce the computational time, we avoid the generation of invalid candidate solutions in the population by incorporating a supervised selection mechanism into the GA operations based on specially designed rules. Specifically, we adopt a constraint-based initialization procedure, and a set of constraint-based operators to generate valid candidate solutions, which is shown in Fig. 1. We use a simple example graph (See Fig.3), which consists of five points and three triangles, to illustrate the process of CBGA labeling.

A Constraint-Based Genetic Algorithm

a) Encoding and Initialization

In this paper, the probability models are encoded using integer strings, and each triangle is typically represented using three integers. Thus, the length of each candidate solution is 3*(*N*-2) for a graph with *N* vertices. In addition, each solution should satisfy the conditions of decomposable graphs. That is to say, when a free vertex is eliminated, the next clique in the ordering will again have a free vertex to eliminate, and so on until the last clique. Here we

develop the rules such that the encoded strings satisfy the conditions. Specifically, the first three integers in the encoded representation are all different, and for the next three integers, one is selected from the unused vertex set, and the other two are selected from the previous triangles. For convenience, we put the new integer in the first position of the three integers which represents the triangle. If these random variables are conditionally independent as described in Fig.3, then:

$$P(ABCDE) = P(ABC)P(D/BC)P(E/BD) \tag{15}$$

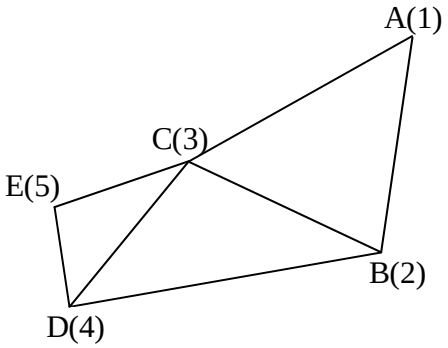

Figure3. The graph of triangle

Based on this encoding scheme, the string "123423524" can be used to represent the individual in Eq. (14). It means that we first randomly generate a set consisting of three vertices ("123" in this case). Then, an additional new vertex ("4" in this case) is generated, and two vertices ("23" in this example) are selected from the previous triangles. This process continues until the last vertex is generated.

We have developed an algorithm to generate valid candidate solutions. Let V denote the set of vertices, V_{use} denote the set of used vertices, V_{unuse} denote the set of unused vertices, and N_{num} denote the length of each candidate solution ($N_{num}/3$ is the number of triangles). V_{use} is initialized as the empty graph, V_{unuse} is initialized as the set V. If $i \le 3$, we sample a vertex $N_i \in V_{unuse}$ and modify V_{use}, V_{unuse} according to Eq. (15) and (16). For the rest of the triangles, we sample a vertex $N_i \in V_{unuse}$ and other two vertices $N_j \in V_{use}$, and modify V_{use} and V_{unuse} as follows.

$$V_{unuse} = V_{unuse} - \{N_i\} \tag{16}$$

$$V_{use} = V_{use} + \{N_i\} \tag{17}$$

We illustrate the initialization procedure in Fig. 3. In the figure, $V = \{1,2,3,4,5\}$. For the first three vertices, we generate three different vertices N_i, for example, $(1,2,3)$. As a result, $V_{unuse} = \{4,5\}, V_{use} = \{1,2,3\}$. Subsequently, we generate three vertices of another triangle (one vertex $N_i \in V_{unuse}$, and two vertices $N_j \in V_{use}$), for example, $(4,3,2)$. Finally, we generate the final three vertices ((5,4,2) in this case). In this way, we generate a valid candidate solution $(1,2,3,4,3,2,5,4,2)$.

b) Fitness Function

The function of the joint mixture Gaussian model is to characterize the relations of the human body parts, and the goodness of fitting to this model is measured by its associated posterior probability. As a result, the joint probability density function of the DTM, which can be automatically learnt from the training data, can be regarded as the objective function to select the candidate solutions during the search processing.

$$f(x) = \log P(X/G) \tag{18}$$

According to the conditional independence of the body joints as represented by the graphical model and the observed human motion sequences $Z = \{Z_1, Z_2, \cdots, Z_J\}$, the joint probability distribution of body joints can be decomposed as,

$$P(Z_i/G) = \prod_{t=1}^{T-1} P(X^i_{A_t}, X^i_{B_t}, X^i_{C_t}) P(X^i_{A_T}, X^i_{B_T}, X^i_{C_T}) \tag{19}$$

$f(x) = \log P(Z/G)$ can be computed as follows,

$$\begin{aligned} f(x) &= \log P(Z/G) \\ &= \sum_{i=1}^{J} \log P(Z_i/G) \\ &= \sum_{i=1}^{J} (\sum_{t=1}^{T-1} \log(X^i_{A_t}, X^i_{B_t}, X^i_{C_t}) + \log P(X^i_{A_T}, X^i_{B_T}, X^i_{C_T})) \end{aligned} \tag{20}$$

Since the $f(x)$ is less than 0, a transformation of the objective function is required so that it can serve as a fitness function in CBGA. Specifically, the following transformation is used:

$$f' = k/(c - f(x)) \tag{21}$$

where *k* is a constant, and $c \geq 0$.This is the expression we used as the fitness function for the GA.

c) Constraint-Based Genetic Operators

The main purpose of the evolutionary operators in GA is to create new valid individuals with higher fitness values in the population. For general GA operations, many iterations are required to find a valid candidate solution. To address this problem, we can adopt suitable measures to avoid the generation of invalid individuals, and to ensure the generation of valid individuals through supervised operations. These two measures can potentially lead to an acceleration of the evolution process.

Selection Operator: The selection of individuals from the current population to form the successive generations plays an important role in GA. In this work, the probabilistic selection based on ranking of the individual's fitness, which was developed by Jonines and Houck [29], is used.

Crossover Operator: To exchange information between different individuals, we generate a uniform random variable N_{rand} within a predefined range, and create a new offspring X by constraint-based exchange. It is the key step to exchange the corresponding triangle. Let V denotes the set of vertices, $V_{X_{huse}}$ the complete set of used vertices for the offspring X ,$V_{X_{cuse}}$ the current set of used vertices for the offspring X , $V_{X_{hunuse}}$ the complete set of unused vertices for offspring X , $V_{X_{cunuse}}$ the current set of unused vertices for offspring X , V_{X_i} the *i*-th position of the offspring X , N_{vector} the number of vertices, and $P_{excchange}$ ($P_{excchange} \in [0,1]$) the crossover probability. $V_{X_{huse}}$ and $V_{X_{cuse}}$ are initialized as empty graphs, and $V_{X_{hunuse}}$, $V_{X_{cunuse}}$ are initialized as the set V . If P_{rand} is less than $P_{excchange}$, where $P_{rand} \in [0,1]$, we perform crossover between two parents $Indvl_A$ and $Indvl_B$ to produce a new offspring. After this operation, we modify the offspring according to,

$$V_{X_i} = \begin{cases} V_{X_i} & \begin{cases} ((i \le 3) \text{ and } (V_{X_i} \in V_{X_{cunuse}})) \text{ or } ((i \bmod 3 = 0) \text{ and } (V_{X_i} \in V_{X_{cuse}}) \text{ and} \\ (V_{X_i} \ne V_{X_{i-1}} \ne V_{X_{i-2}})) \text{ or } ((i \bmod 3 = 1) \text{ and } (V_{X_i} \in V_{X_{cunuse}})) \mid \\ ((i \bmod 3 = 2) \text{ and } (V_{X_i} \in V_{X_{cuse}}) \text{ and } (V_{X_i} \ne V_{X_{i-1}})) \end{cases} \\ \mathrm{V}_{X_{rand}}, \mathrm{V}_{X_{rand}} \in V_{X_{hunuse}} & (i \le 3) \text{ or } (i \bmod 3 = 1) \\ \mathrm{V}_{X_{rand}}, \mathrm{V}_{X_{rand}} \in V_{X_{cuse}} & otherwise \end{cases} \tag{22}$$

$$V_{X_{cuse}} = \begin{cases} V_{X_{cuse}} + \{V_{X_i}\} & (i \le 3) \text{ or } (i \bmod 3 = 1) \\ V_{X_{cuse}} & otherwise \end{cases} \tag{23}$$

$$V_{X_{cunuse}} = V - V_{X_{cuse}} \tag{24}$$

$$V_{X_{huse}} = \begin{cases} V_{X_{huse}} + \{V_{X_i}\} & V_{X_i} \in V_{X_{hunuse}} \\ V_{X_{huse}} & otherwise \end{cases} \tag{25}$$

$$V_{X_{hunuse}} = V - V_{X_{huse}} \tag{26}$$

We provide an example here to illustrate the constraint-based crossover operator. Given two parents $Indvl_A$ and $Indvl_B$,

$$Indvl_A = (5\ 4\ 3\ 2\ 3\ 4\ 1\ 2\ 3)$$

$$Indvl_B = (1\ 2\ 4\ 3\ 4\ 2\ 5\ 1\ 2)$$

Suppose $P_{rand} < P_{exchange}$, we perform crossover between the two parents to obtain the following offspring.

$$offspring = (5\ 4\ \underline{4}\ \underline{3\ 4\ 2}\ \underline{5\ 1\ 2})$$

However, the offspring we find may not satisfy the constraints. For example, if the vertices in the first position and the third position are the same, it cannot form a triangle. Therefore, it is necessary to modify the offspring according to the constraints. Initially, we obtain the set of $V_{X_{huse}}$ and $V_{X_{hunuse}}$ as follows,

$$V_{X_{huse}} = \{5\ 4\ 3\},\ V_{X_{hunuse}} = \{1\ 2\}$$

Then, we modify the set of $V_{X_{cuse}}$ and $V_{X_{cunuse}}$ as follows.

$$V_{X_{cuse}} = \{5\ 4\},\ V_{X_{cunuse}} = \{1\ 2\ 3\}$$

We then continue the constraint checking process and perform the following adjustments:

$$offspring = (5\ 4\ \underline{2}\ 3\ 4\ 2\ 5\ 1\ 2)$$

$$V_{X_{huse}} = \{5\ 4\ 3\ 2\}$$

$$V_{X_{hunuse}} = \{1\}$$

$$V_{X_{cuse}} = \{5\ 4\ 2\}$$

$$V_{X_{cunuse}} = \{1\ 3\}$$

For the second triangle, we find that it satisfies the constraints, so just modify the set of $V_{X_{huse}}$, $V_{X_{hunuse}}$, $V_{X_{cuse}}$ and $V_{X_{cunuse}}$.

$$V_{X_{huse}} = \{5\ 4\ 3\ 2\}$$

$$V_{X_{hunuse}} = \{1\}$$

$$V_{X_{cuse}} = \{5\ 4\ 3\ 2\}$$

$$V_{X_{cunuse}} = \{1\}$$

For the last three vertices, we find that the in the seventh position $V_{X_7} \in V_{V_{cuse}}$ ($V_{X_7} = (5)$), the constraint is not satisfied. As a result, we generate a new value $V_{X_7} \in V_{hunuse}$ as follows,

$$offspring = (5\ 4\ 2\ 3\ 4\ 2\ \underline{1}\ 1\ 2)$$

However, the last three numbers $(1\ 1\ 2)$ does not satisfy the triangle constraints. Thus, we adjust the eighth position ($V_{X_S} \in V_{X_{cusse}}$) until they satisfy the conditions.

$$offspring = (5\ 4\ 2\ 3\ 4\ 2\ 1\ \underline{3}\ 2)$$

Mutation Operator: The constraint-based mutation operator is applied by two steps. The first step is that a selected element in the individual is replaced by a random value produced from the predefined range by a small probability. Then, these constraints are applied to produce a valid individual. Let V denote the vertices, V_{use} denote the set of used vertices, V_{unuse} denote the set of unused vertices, V_i denote the chromosome of an individual $Indvl$, N_{vector} denote the number of vertices, $P_{mutation}$ ($P_{mutation} \in [0,1]$) denote mutation probability. The

initial value for V_{use} is an empty graph, and the initial value for V_{unuse} is the set V. After mutation operator and producing a random variable Num_{rand}, the constraints are used.

The constraint-based mutation operator is applied in two steps: In the first step, a selected element in the individual is replaced by a random value sampled from a predefined range. Then, constraints are applied to produce a valid individual. Let V denote the set of vertices, $V_{X_{use}}$ the set of used vertices, $V_{X_{unuse}}$ the set of unused vertices, V_{X_i} the representation of an individual $Indvl$ in the population, N_{vector} the number of vertices, and $P_{mutation}$ ($P_{mutation} \in [0,1]$) the mutation probability. $V_{X_{use}}$ is initialized as the empty graph, and $V_{X_{unuse}}$ is initialized as the set V. After applying a mutation operator based on a random variable $V_{X_{rand}}$, the constraints are enforced.

$$V_{X_i} = \begin{cases} V_{X_{rand}} & \begin{cases} ((i \le 3) \text{ and } (V_{X_{rand}} \in V_{X_{unuse}})) \text{ or } ((i \bmod 3 = 0) \text{ and } (V_{X_{rand}} \in V_{X_{use}}) \text{ and} \\ (V_{X_{rand}} \ne V_{X_{i-1}} \ne V_{X_{i-2}})) \text{ or } ((i \bmod 3 = 1) \text{ and } (V_{X_{rand}} \in V_{X_{unuse}})) \text{ or} \\ ((i \bmod 3 = 2) \text{ and } (V_{X_{rand}} \in V_{X_{use}}) \text{ and } (V_{X_{rand}} \ne V_{X_{i-1}})) \end{cases} \\ reproduce V_{X_{rand}} & otherwise \end{cases} \tag{27}$$

It is necessary to modify the set of $V_{X_{unuse}}$ and $V_{X_{use}}$ after the mutation operator is applied.

$$V_{X_{use}} = \begin{cases} V_{X_{use}} + \{V_{X_i}\} & (i \le 3) \text{ or } (i \bmod 3 = 1) \\ V_{X_{use}} & otherwise \end{cases} \tag{28}$$

$$V_{X_{unuse}} = \begin{cases} V_{X_{unuse}} - \{V_{X_i}\} & (i \le 3) \text{ or } (i \bmod 3 = 1) \\ V_{X_{unuse}} & otherwise \end{cases} \tag{29}$$

The following example illustrates the procedure of constraint-based mutation. Given a parent $Indvl$,

$$Indvl = (5\ \ 4\ \ 3\ \ 2\ \ 2\ \ 4\ \ 1\ \ 2\ \ 3)$$

For the selected element, we generate a random variable P_{rand}. We will modify the element if $P_{rand} < P_{mutation}$, and make it satisfy the constraints. Suppose we select the third position and $P_{rand} < P_{mutation}$. Then,

$$V_{X_{use}} = \{5\ \ 4\}, \ V_{X_{unuse}} = \{1\ \ 2\ \ 3\}$$

According to the rules, we replace this element with another randomly chosen element $V_{X_3} \in V_{X_{unuse}}$, and remove V_{X_3} from $V_{X_{unuse}}$ and add V_{X_3} to $V_{X_{use}}$.

The mutation may produce some individuals which do not satisfy the constraint. Thus, it is necessary to check these new individuals. Finally, the genetic algorithm will be terminated once a predefined Maximum number of generations are reached, or the value of the fitness function does not change over a large number of iterations.

B. Handling Missing Body Parts

We have shown how to label the body parts if they are observable. When some parts are occluded, the corresponding measurements for these body parts will be missing. In such cases the missing parts can be modeled as additional hidden variables [30], and CBGA can be modified to handle this problem using the probability estimation similar to [27]. Specifically, the foreground probability density function is the marginalized version of the above equations, and it is necessary to modify the fitness function accordingly. For example, in Fig.3, if A is missing and the conditional independence as in Eq. (4) holds, then the PDF is,

$$\begin{aligned} P(A,B,C,D,E) &= P(B,C,D,E) \\ &= P(B,C)P(D/B,C)P(E/C,D) \end{aligned} \tag{30}$$

If D is missing, then,

$$\begin{aligned} P(A,B,C,D,E) &= P(A,B,C,E) \\ &= P(A,B,C)P(E/C) \end{aligned} \tag{31}$$

Thus, we can calculate the fitness value according to the above equations and use the CBGA to search for the optimal results of the using the modified fitness function. Note that if too many body parts are missing, the conditional independence assumption of the graphical model will no longer hold.

C. Integrating Information from Multiple Frame

So far, we have only assumed that we use information from two consecutive frames, from which we obtain the positions and velocities as features. In this section we will like to extend our previous results to the case where multiple frames are available, and we adopt a voting mechanism in which $N_{Num}(i,k)$ denotes the vote for the k-th vertex in the i-th candidate solution, and P_{ij} the j-th position in the i-th solution. The $N_{Num}(i,k)$ are initialized as zeros,

and these votes are incremented for the different vertices based on information from the different frames. We then perform labeling for the vertices based on the maximum number of votes received as follows:

$$N_{Num}(i, P_{ij}) = N_{Num}(i, P_{ij}) + 1 \tag{32}$$

V. Experimental Results

In this section we perform experiments to evaluate the performance of our proposed approach. Our algorithm is applied to motion capture data sets obtained from the CMU Graphics lab [31], which has 42 articulations with 3D coordinates. In the experiments, we select 14 articulations (see Fig.2). In the following sections we will use four sequences (walking sequences W1 and W2, running sequences R1 and dancing sequences D1 in Fig.4~Fig.7) to perform training and testing.

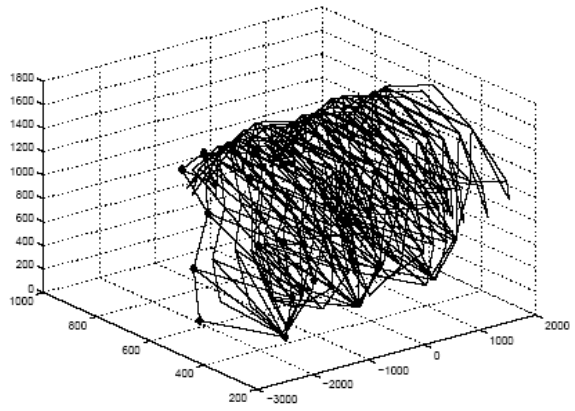

Figure4. Walking Sequence 1

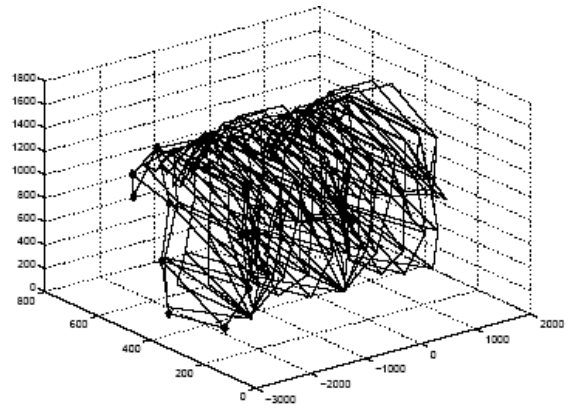

Figure5. Walking Sequence 2

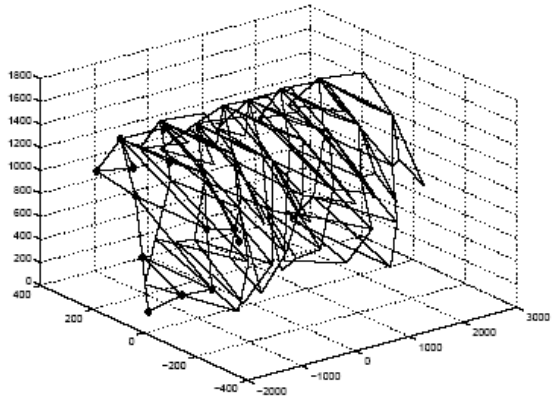

Figure6. Running Sequence

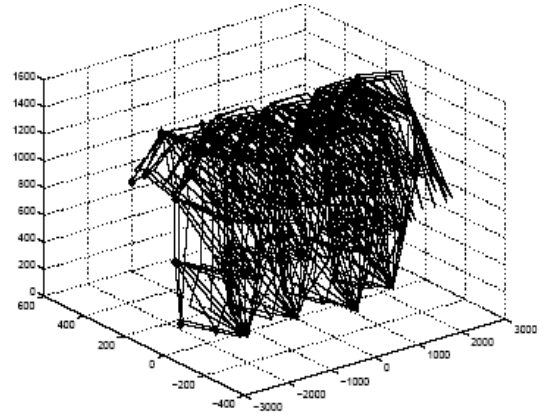

Figure7. Dancing Sequence

A Evaluation of the Probably Models

In this section, we adopt the structure shown in Fig.2 for our experiments. For each sequence, we train tree models based on the following distributions: joint Gaussian, joint mixture Gaussian with fixed number of components, and joint mixture Gaussian with different number of components. Overall, we obtain 12 different models for every distribution.

At the test phase, we use two approaches: in the first approach, we select the model corresponding to the maximum probability from 12 different models of a given triangle; in the second approach, we select the triangle with the maximum probability from 12 different triangles for the given model. Ideally, the correct combination of markers should produce the highest probability for the given model, and the corresponding model should also attain the highest probability for the given model. Otherwise, an error occurs. In the Gaussian mixture models with a fixed number of components, the corresponding number is 3.The ranges for the number of Gaussian mixture components are set as $K_{\min} = 1 \text{and} K_{\max} = 8$.

Fig.8 shows the accuracy rates for each human motion class using joint Gaussian, joint mixture Gaussian and joint mixture Gaussian with different numbers of components probability models using the first approach respectively. From this figure, it can be seen that the results for the joint Gaussian mixture distribution with different number of components are superior to those corresponding to using a joint probability distribution of a single Gaussian, and the joint Gaussian mixture distribution with a constant number of components, especially for the 10th, 11th and 12th triangles which correspond to the knee. The average results for different sequences have been shown in Tab.1.

To further investigate the performance of the mixture Gaussian model with different number of components. We adopt the second approach to detect every triangle in Fig.2. Fig.9 shows the results for detection of the given triangles using the joint Gaussian, mixtures Gaussian and mixtures Gaussian with different number of components. It can be seen that the mixtures of Gaussian with different number of components performs better than the other two probabilistic models for each triangle again, especially for the 9th, 10th, 11th and 12th triangle. We obtained 100% correct rate for mixtures Gaussian with different number of components, 15% to 96% for a single Gaussian, and 60% to 100% for mixtures of Gaussian with a constant number of components. The average results for the different sequences are shown in Tab.2, where JG, JMG and JMGW denote joint Gaussian model, joint mixture Gaussian model and joint mixture Gaussian model with different number of components respectively.

Table 1: Correct Rates to test the probabilistic model by the first method

Methods	Walking1	Walking2	Running	Dancing
JG	85.43%	75.84%	76.10%	85.01%
JMG	100%	98.87%	99.58%	99.62%
JMGW	100%	100%	100%	100%

Table 2: Correct Rates to test the probabilistic model by the first method

Methods	Walking1	Walking2	Running	Dancing
JG	62.12%	77.17%	79.53%	78.79%
JMG	99.96%	99.12%	99.51%	99.08%
JMGW	100%	100%	100%	100%

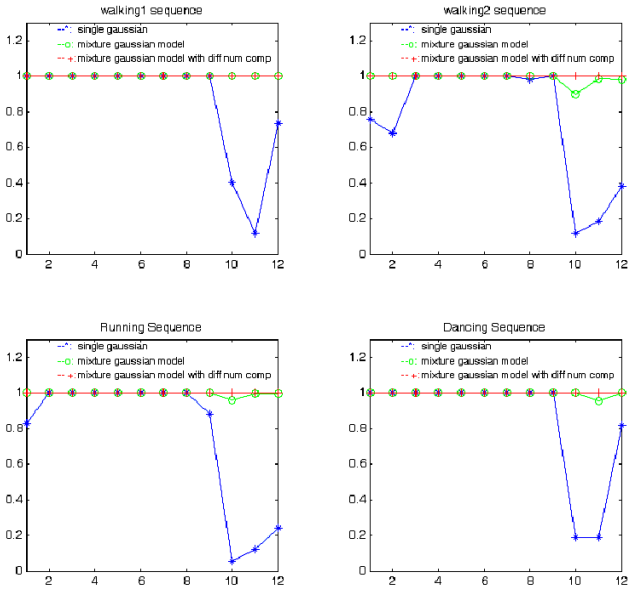

Figure8. The accuracy rates by the first method using single Gaussian, mixture Gaussian model and mixture Gaussian model with different number of components for the two walking sequences, running sequence and dancing sequence. The red, green and blue color represent mixture Gaussian model with different number of components, mixture Gaussian model and single Gaussian model, respectively.

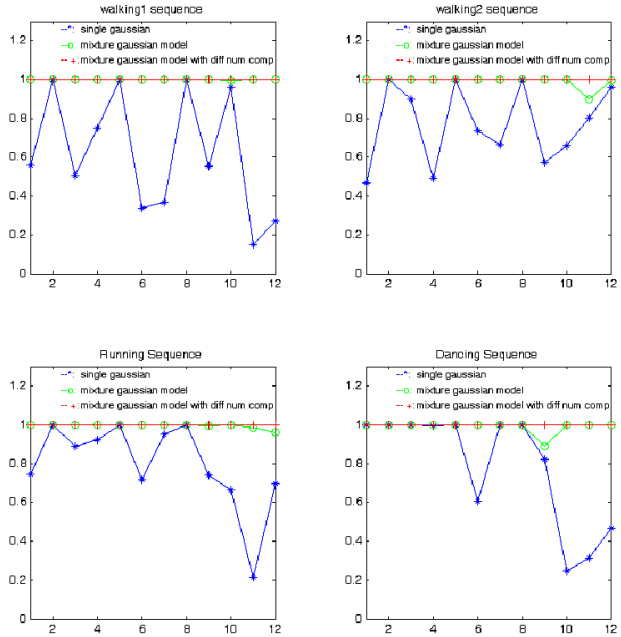

Figure9. The accuracy rates by the second method using single Gaussian, mixture Gaussian model and mixture Gaussian model with different number of components for the two walking sequences, running sequence and dancing sequence. The red, green and blue color represent mixture Gaussian model with different number of components, mixture Gaussian model and single Gaussian model, respectively.

B. Performance of CBGA

For the purpose of comparison, simple GA (SGA) optimization is applied to the four

sequences. We compare the number of iterations and the accuracy rates by CBGA and SGA with mixture Gaussian model with different number of components using the current human model (See Fig.2). Each training process uses 9/10 of the entire data set, with the remaining 1/10 data records used for testing, and training is terminated when a pre-defined number of generations is reached. Both the training and testing results of the SGA and CBGA using the same data set are summarized in Fig.11 and Table.3, which are based on cross-validation. Fig.10 shows the iteration times in the four sequences.

The performance of SGA ranges from 87.61% to 94.27%, while that of CBGA ranges from 89.23% to 96.29%. The evolution of the fitness value is shown in Fig.10. The number of generations for CBGA varies from 50 to 100, while SGA converges to the maximum fitness value after 500 generations for each sequence.

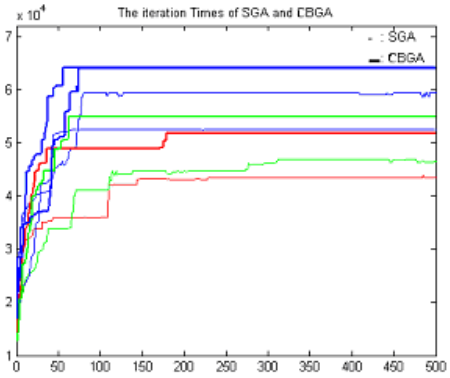

Figure10. The fitness function values of SGA and CBGA vs. different iteration times. The red, green and blue curves represent the Dancing, Running and Walking Sequences, respectively. X-axis is the iteration times, Y-axis is the fitness value.

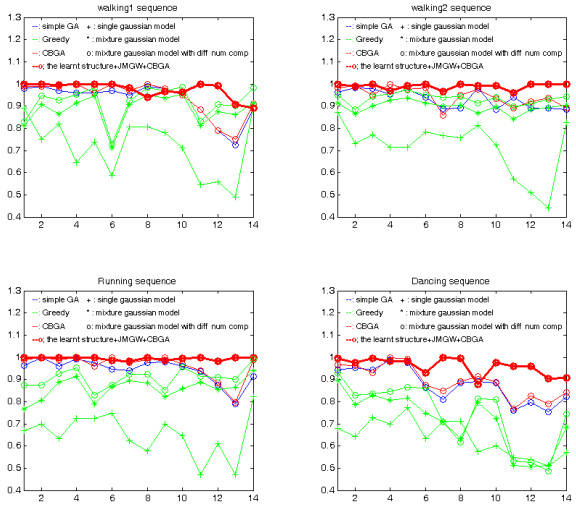

Figure11 Labeling Results using different methods. The red, green and blue color represent

mixture Gaussian model with different number of components, mixture Gaussian model and single Gaussian model, respectively. The coarse red line shows the results using the learnt model, CBGA and mixture Gaussian with different number of component.

C. Labeling Performance

To provide a better comparison between CBGA and the greedy algorithm, we test the labeling performance on the four sequences using the given human model (See Fig.2), and the learnt human model (See Fig.12), which are derived from CBGA. Each sequence is divided into ten segments, and frames from nine of the segments are used as the training set. The average accuracy rates for the different sequences and the accuracy rates for the different articulations are shown in Tab.3 and Fig.11. It can be seen that the error rates of the last four markers are higher than the other markers, which is natural since these triangles correspond to the left and right knees, and they are so close in some of the frames that it is even hard for human eyes to distinguish between them. The average results for all the sequences are at least 89.2% when CBGA is applied to the given model. Especially, a 96.3% accuracy rate is attained in the case of the running sequence using CBGA.

We also evaluate the performance of the joint Gaussian model, joint mixture Gaussian model and joint mixture Gaussian model with different number of components using greedy optimization. Compared to the previous accuracy rates of CBGA, the performance of the greedy algorithm ranges from 74.27% to 93.36% when mixture Gaussian with different number of components is used, from 71.14% to 89.64% when mixture Gaussian is used, and from 6% to 93.36% when a joint Gaussian is used. It can be observed from these results that the labeling performance based on the adoption of a mixture Gaussian model in CBGA is in general better than those based on the greedy algorithm using the joint Gaussian and the mixture Gaussian probabilistic models. After obtaining the structures for the four sequences, as shown in Fig.12, we then identify the body features based on the optimized structures.

We also evaluate our approach in the case of occlusion by randomly removing points or choosing frames, and the results are shown in Fig.13 for different motion sequences. The corrected rate is more than 80% when less than three points are occluded for the all sequences.

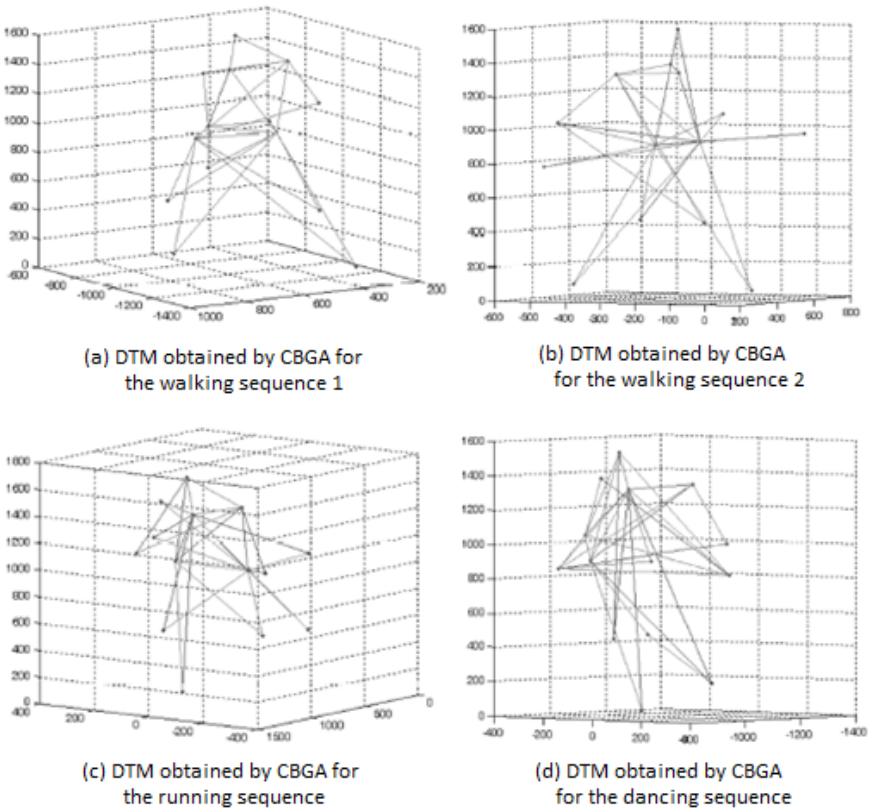

Figure12. DTM obtained by CBGA for the four sequences

Table 3: Correct Rates to test the probabilistic model by the first method

Methods	Walking1	Walking2	Running	Dancing
Gassian+Greedy	71.66%	71.4%	65.1%	63.7%
MixG+Greedy	88.36%	89.64%	86.05%	71.14%
MixGM+Greedy	91.71%	93.36%	90.76%	74.27%
MixGM+SGA	92.71%	93.14%	94.27%	87.61%
MixGM+CBGA	93.98%	94.56%	96.29%	89.23%
MixGM+CBGA+Struct	97.4%	99.01%	99.48%	95.95%

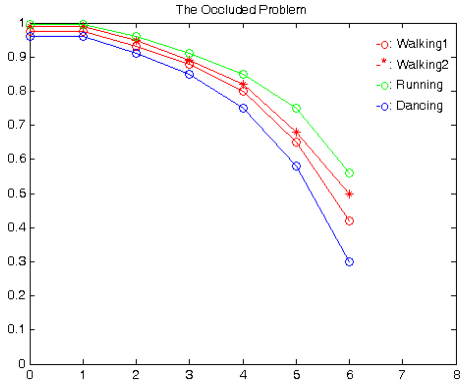

Figure13. Correct Rates vs. the number of occluded body parts

In the previous experiments, the data used are acquired through an accurate motion capture system. To evaluate the performance of our proposed approach when measurement noise is present, we add independent Gaussian noise to the position of the joints. Fig.14 shows the correct labeling rate when Gaussian noise is added to positions for W1 sequence. The red, blue and green curves represent the labeling results by the multi-Gaussian model using CBGA, multi-Gaussian model using greedy algorithm and single-Gaussian model using greedy algorithm respectively. From the results, we can see that our method provides a higher accuracy rate in labeling, and it is more robust than other approaches.

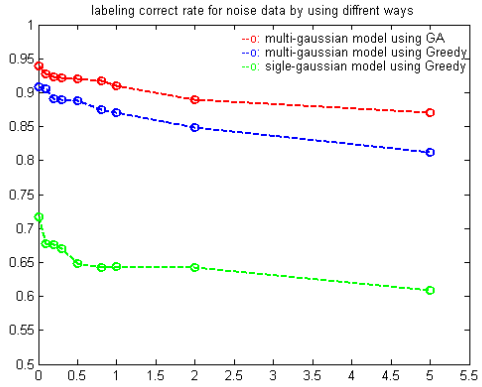

Figure14. Correct labeling rate vs. standard deviation

D. Using information from multiple frames

We can further improve the correct rate of labeling by integrating information from multiple frames and using the voting system described in section 4.2. Fig.15 plots the detection rate vs. the number of integrated frames. The detection rate is almost 100% when more than 5 frames are used for the all sequences.

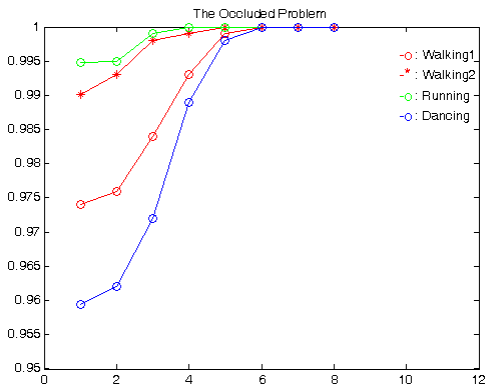

Figure15. Detection rate vs. number of frames used

VI. CONCLUSION

In this paper, we propose a human feature labeling approach in which constraint-based genetic algorithm is used to search for the optimal model, according to which labeling is performed. Specifically, the human body structure is represented as a decomposable triangulated graph, with each triangle associated with a joint mixture Gaussian distribution with different number of components. We have applied this approach to optimize the graph model and perform labeling of the body features. Our proposed approach is also capable of handling occlusion, and the results can be further improved by integrating information from multiple frames. In addition, the adoption of the joint mixture Gaussian with different number of components is able to further improve the labeling accuracy. The proposed CBGA is generic and problem independent, and allows the inclusion of domain knowledge to reduce the search space during the evolutionary optimization process. Compared with simple GA and greedy optimization approaches, the model optimized through CBGA provides higher labeling accuracy rates, which are attained through a fewer number of generations.

ACKNOWLEDGMENT

This research was supported by a grant from City University of Hong Kong (Project No. 7001766), Nature Foundation of Jiangsu Province (Project No. BK2012166), Natural Science Foundation of Jiangsu University (Project No.12KJB10031), Construction system of science and technology project of Jiangsu Province(Project No. JH21) and Open Foundation of Modern enterprise information application supporting software engineering technology R&D center of Jiangsu Province (SK201206).

REFERENCES

[1] D. Cosker, E. Krumhuber, and A. Hilton, “A FACS valid 3D dynamic action unit database with applications to 3D dynamic morphable facial modeling”, Proceeding of IEEE International Conference on Computer Vision, Barcelona, Spain, pp.2296-2303, 2011.

[2] P. Matikainen, R. Sukthankar and M.Hebert, “Feature seeding for action recognition”, Proceeding of IEEE International Conference on Computer Vision, Barcelona, Spain, pp.1716-1723, 2011.

[3] H. Rimminen, J. Lindstrom and R. Sepponen, “Positioning accuracy and multi-target

separation with a human tracking system using near field imaging", International Journal on Smart Sensing and Intelligent Systems, vol.2, no.1, pp.156-175, 2009.

[4] G. Kim, E.P. Xing and A. Torralba, "modeling and analysis of dynamic behaviors of web image collections", Proceeding of European Conference on Computer Vision, Heraklion, Crete, Greece, pp.85-98, 2010.

[5] J. Y. Lee and K. Yao, "Multi-target, multi-sensor tracking based on quality-of-information and formal bayesian frameworks", International Journal on Smart Sensing and Intelligent Systems, vo1.1, no.4, pp.842-851, 2008.

[6] X. Wang, M.Q. Li, S. Liu and Q. Ma, "A vision-based real time motion synthesis system", Proceeding of International Symposium on Computational Intelligence and Design, Zhejiang Sci-Tech University, China, pp.548-553, 2012.

[7] Y. Li, T. S. Wang and H.Y. Shum, "Motion Texture: A Two-level Statistical Model for Character Motion Synthesis", Proceeding of ACM SIGGRAPH, pp.465-472, 2002.

[8] P. Frank, M. Atsuto and S. Bjorn, "Correlated probabilistic trajectories for pedestrian motion detection", Proceeding of IEEE International Conference on Computer Vision, Kyoto, Japan, pp.1647-1654, 2009.

[9] P. Huang, A. Hilton and J. Starck, "Human motion synthesis from 3D video", Proceeding of IEEE Conference on Computer Vision and Pattern Recognition, pp.1478-1485, 2009.

[10] M.C. Chang, N. Krahnstoever and W. Ge, "Probabilistic group-level motion analysis and scenario recognition", Proceeding of IEEE International Conference on Computer Vision, Barcelona, Spain, pp.747-754, 2011.

[11] S. K. M. Wong and F.C.S. Poon, "Comments on Approximating Discrete Probability Distributions with Dependence Trees", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.11, no.3, pp.333-335, 1989.

[12] B. Sapp, C. Jordan, B. Taskar, "Adaptive pose priors for pictorial structures", Proceeding of IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, pp.422-429, 2010.

[13] A. Torsello and E.R. Hancock, "Learning shape-classes using a mixture of treeunions", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.28, no.6, pp.954-967, 2006.

[14] S. Nathan, "Maximum likelihood bounded tree-width markov networks", Proceeding of UAI, San Francisco, CA, pp.504-511, 2001.

[15] R. Li, T.P. Tian and M.H. Yang, "3D human motion tracking with a coordinated mixture of factor analyzers", International Journal of Computer Vision, vol.87, no.1-2, pp.170-190, 2010.

[16] Y. Amit and A. Kong, "Graphical templates for model registration", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.18, no.3, pp.225-236, 1996.

[17] M. Vondrak, L. Sigal and O.C. Jenkins, "Physical simulation for probabilistic motion tracking", Proceeding of IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, Alaska, USA, pp.1-8, 2008.

[18] S. Yang, X. Feng and P. Perona, "Towards detection of human motion", Proceeding of IEEE Conference on Computer Vision and Pattern Recognition, Hilton Head, SC, USA, pp.810-817, 2000.

[19] Y. Song, L. Goncalves, E. Di Bernard and P. Perona, "Monocular Perception of Biological Motion in Johansson Displays", Computer Vision and Image Understanding, vol.81, no.3, pp. 303-327, 2001.

[20] S. Yang, L. Goncalves and P. Perona, "Unsupervised Learning of Human Motion", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.25, no.7, pp.814-827, 2003.

[21] P. Larranga, C. M. H. Kuijpers, R. H. Murga and Y. Yurrramendi, "Learning Bayesian network structure by searching for the best ordering with genetic algorithms", IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, pp.487-193, 1996.

[22] M.Garofalakis and R. Rastogi, "Scalable data mining with model constraints", SIGKDD Explorations, pp.39-48, 2000.

[23] C. C. Chiu and P. L. Hus, "A Constraint-Based Genetic Algorithm Approach for Mining Classification Rules", IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol.35, no.2, pp.205-220, 2005.

[24] N. Barnier and P. Brisset, "Optimization by hybridization of a genetic algorithm and constraint satisfaction techniques", Proceeding of IEEE World Congress Computational Intelligence - Evolutionary Computation, pp.645-649, 1998.

[25] N. Bouguila and D. Ziou, "MML-Based Approach for High-Dimensional Unsupervised Learning Using the Generalized Dirichlet Mixture", Proceeding of IEEE Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, pp.53-60, 2005.

[26] Z. Zivkovic and F. van der Heijden, "Recursive unsupervised learning of finite mixture models", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.26, no.5, pp.651-656, 2004.

[27] S. Pierard, A. Lejeune and M. Van Droogenbroeck, "A probabilistic pixel-based approach to detect humans in video streams", IEEE International Conference on Acoustics, Speech and Signal Processing, pp.921-924, 2011.

[28] F.M.J. Willems, "Elements of Information Theory", IEEE Trans on Information Theory, vol.39, no.1, pp.313-315, 1993.

[29] J.A. Joines and C.R. Houchk, "On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GA's", In Proc. of IEEE Conference on Evolutionary Computation, vol.2, pp.579-584, 1994.

[30] H.L. Trung, V.L. Anh and K.L. Trung, "An unsupervised learning and statistical approach for Vietnamese word recognition and segmentation", Proceeding of Asian Conference on Intelligent Information and Database System, pp.195-204, 2010.

[31] Carnegie Mellon University Graphics Lab Motion Capture Database, http://www.mocap.cs.cmu.edu.